SYNOPSIS REPORT ON

DEVNAGARI SIGN LANGUAGE RECOGNITION USING MACHINE LEARNING

Submitted to the

MGM University

Chhatrapati Sambhajinagar (MS) - 431003, INDIA

For the Award of the Degree

DOCTOR OF PHILOSOPHY

In

COMPUTER SCIENCE & ENGINEERING

By

Deepali Rajaram Naglot

(PH21CSE12)

Under Guidance of

Dr. Deepa Deshpande

Department of Computer Science and Engineering,
Jawaharlal Nehru Engineering College (JNEC)
MGM University

April 2025

DECLARATION

I hereby declare that, the present work completed in the form of synopsis entitled, "Devnagari Sign Language Recognition Using Machine Learning" is an original work carried out by me under the guidance of Dr. Deepa Deshpande, and has not been submitted or published in any form for the fulfilment of any other degree or any other similar to this in other University.

I further declare that the material obtained from other sources has been duly acknowledged in the synopsis.

Deepali R. Naglot

RESEARCH SCHOLAR NAME AND SIGN

PLACE: CHHATRAPATI SAMBHAJINAGAR

DATE:

MGM University

Chhatrapati Sambhajinagar

CERTIFICATE

This is to certify that Deepali R. Naglot has carried out the work embodied in this synopsis entitled "**Devnagari Sign Language Recognition Using Machine Learning**". The work included in this synopsis is original, unless stated otherwise and has not been submitted for other degree of MGM University or any other University. References made to the work of others have been cited in the text.

SIGNATURE AND NAME OF RESEARCH SUPERVISOR

INDEX

Particulars	Page No
Introduction	1
Summary of Literature Survey and Research Gap	2
Research Objectives and Scope	3
Contributions	4
Proposed Methodology and Results	5-16
Conclusion	17
Organisation of the Thesis	18
References	19
List of Publications and Conference Attended	21

INTRODUCTION

According to the World Health Organization, hearing and speaking impairment affects nearly 5% of the global population, with approximately 63 million individuals in India experiencing aural and verbal disabilities. Individuals with hearing and speech impairments mainly depend on sign language to communicate; however, this can pose difficulties for those who are not well-versed in it [1]. Developing efficient solutions to promote effective exchange of information between those with hearing loss and the wider society is crucial for bridging this communication gap.

Although areas have been conducted since the early 1990s, advancements have remained relatively modest, primarily due to challenges in sensor and computing technology. Recognizing real-time sign language remains technically demanding due to challenges such as varying lighting conditions, intricate backgrounds, diverse hand positions, camera angles, occlusion, and inconsistencies in sign production. While investigators have researched different national sign languages, there has been limited attention to Devnagari Sign Language (DSL), which is widely used for communication in India. DSL is rooted in the Devnagari script, which serves as the writing system for languages such as Hindi, Marathi, Nepali, and Sanskrit in South Asia. The Fig.1 shows sample of the Devnagari Sign Language alphabets from (Aa) to (Dnya)[2]. Developing a DSL interpretation system can significantly facilitate better communication for those with auditory and verbal challenge in India. In Devnagari, vowels (Swar) are referred to as ‘Svr,’ signifying articulation or vocalization. The script contains 34 consonants, ranging from ‘क’ to ‘ळ,’ and forms the foundation for writing Marathi and Hindi. The combination of vowels and consonants in Devnagari is known as "Barakhadi," encompassing not only the core characters but also specific symbols influenced by linguistic and cultural traditions [3].

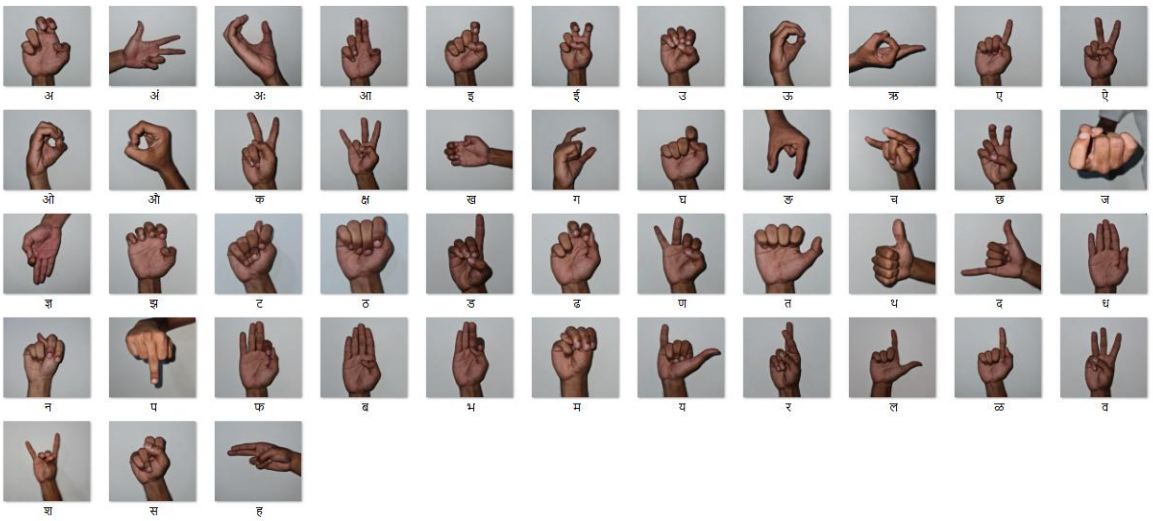

Fig 1. Devnagari Sign Language Alphabets

Proposed system developed two models: Image-based and Hand Landmark based DSL recognition. Image-based DSL recognition system includes data collection, augmentation, pre-processing, feature selection and classification. System used pre-trained CNN models such as InceptionV3, ResNet50, VGG16, AlexNet and Customized CNN For feature selection & classifications. Hand Landmark based DSL recognition includes data collection, image augmentation followed by hand landmark collection, classification using SVM, FFNN, random forest, and customized LSTM-RNN, and model testing. This model worked on 47 Devnagari alphabets and 408 Barakhadi alphabets.

SUMMARY OF LITERATURE SURVEY AND RESEARCH GAP

A detailed examination of existing research was undertaken to identify gaps in existing Sign Language Recognition System. Many researchers have worked on sign languages of their respective countries, including Arabic Sign Language (ArSL), American Sign Language(ASL),Chinese Sign Language(CSL), Bhutanese Sign Language (BSL), Indian Sign Language (ISL) and others. However, limited attention has been given to Devnagari Sign Language. In the existing system, dataset is collected using a web camera, Microsoft Kinect Camera, and Leap Motion Controller. Researchers have been using a different method to extract discriminative features and create powerful classifier. Following are research gaps observed in the existing system which handled in the proposed research work.

Research Gap

1. Limited work on Devnagari Sign Language.
2. Background constraints and Size of the dataset.
3. Failure in identification of similar signs.
4. Most of the research work focuses on digits only and few alphabets of DSL.
5. None of the researchers works on Devnagari Sign Language Barakhadi recognition.

RESEARCH OBJECTIVES AND SCOPE

Objectives of Research

This research primarily seeks to develop a machine learning based System for recognizing Devnagari sign language gestures. It aims to:

1. Bridge communication barrier between normal and deaf & dumb person.
2. Give standard platform to deaf and dumb people for education, social inclusion, and work environments.
3. Develop a computer aided system using machine learning approach for recognition and interpretation of Devnagari sign language
4. Evaluate the performance of various CNN architectures for image-based recognition of Devnagari Sign Language alphabets.
5. Compare different machine learning classifiers for hand landmark-based recognition of Devnagari Sign Language alphabets.
6. Recognize Barakhadi in Devnagari Sign Language using machine learning approaches.

Scope

1. Investigating current machine learning algorithms and deep learning models for recognizing DSL alphabets
2. Designing and implementing a system that can classify Barakhadi alphabets of Devnagari Sign Language
3. Analyzing the performance of the system compared to existing solutions in terms of detection accuracy, adaptability, and scalability.

CONTRIBUTIONS

***Contributions*:** This research's significant contributions are listed below:

1. A diverse dataset has been curated, encompassing individuals of different ages and genders, which enables detailed analysis and ensures the reliability of the resulting insights.
2. The research conducts a detailed examination and comparison of the efficiencies exhibited by various pre-trained model such as InceptionV3, ResNet50, AlexNet, VGG16 and Customized Convolutional Neural Network (CNN) model, We found that performance of **InceptionV3** is better than other model and **achieves higher accuracy.** By systematically evaluating these models, the study sheds light on their respective strengths and weaknesses, thus offering insights for the future research.
3. Image augmentation with hand tracking, demonstrating reliable performance across various environmental conditions. We compare different classifiers, including SVM, Random Forest, FFNN, and LSTM-based RNN, and show that the **LSTM with Attention Mechanism** outperforms the others, achieving higher accuracy in detecting DSL alphabets in both static and real-time modes.
4. The research contributes by demonstrating the combination of vowel signs with consonants to form the full Barakhadi of the Devnagari script, resulting in a comprehensive set of 408 characters used as class labels.
5. The research introduces a robust classification methodology specifically tailored for Barakhadi recognition of Devnagari sign language.

PROPOSED METHODOLOGY

1. Module 1: Image-based DSL Recognition System

Dataset are captured from a range of people for the collection of Devnagari alphabet sign using mobile phone camera. The dataset features signers of both sexes across multiple age groups. Each Devnagari sign is recorded against a consistent background and under optimal lighting. Techniques for data augmentation are utilized to mitigate over fitting, thereby enhancing the resilience and adaptability of the model. Proposed system Pre-trained models InceptionV3, ResNet50, VGG16, and AlexNet are selected based on their performance. Moreover, a tailored three-layered CNN architecture is designed, trained from the ground up, and compared with other models. The objective is to evaluate the deep learning models, optimization strategies, and parameter configurations concerning Devnagari Sign Language. The evaluation process follows steps as shown in Fig 2.

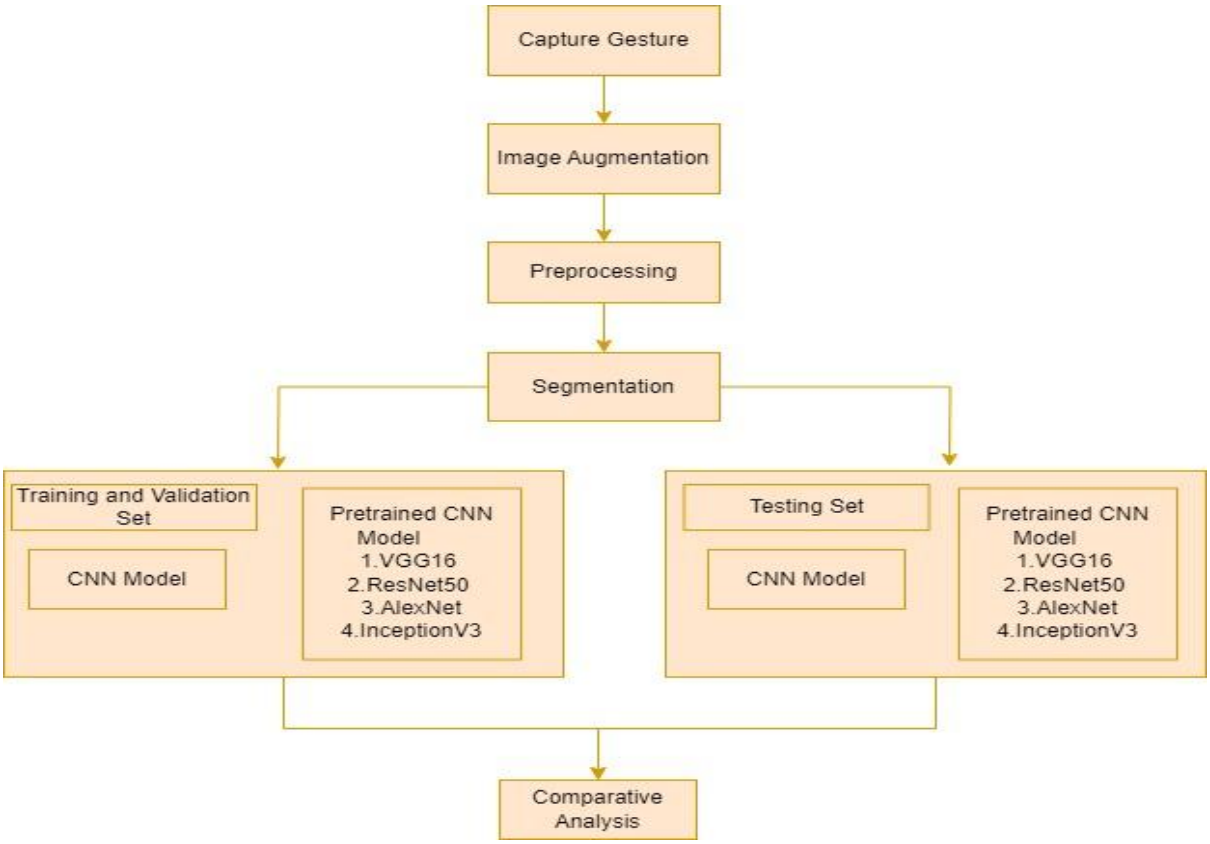

Fig 2. Architecture of Image-based DSL Recognition System

1.1. Dataset Collection:

Dataset collected from both sexes across multiple age groups in unique background. Image augmentation techniques such as Scaling, Rotation and Shear transformations applied to enhance dataset. The Total dataset comprised 4230 images distributed across 47 classes as 90 images of each alphabet [4].

1.2. Hand Segmentation:

The image is converted to YCrCb color space by defining the lower threshold of (0, 138, 67) and upper threshold of (255, 173, 133) for the skin color in the YCrCb color space. After that, Pixels within the specified skin color range are set to white (255), and pixels outside the range are set to black (0). Applied morphological operations to improve the mask by removing small noise regions and smoothens the edges using an opening operation and fill in small holes within the skin region

using a closing operation. Finally, the refined mask is applied to the original image, resulting in the segmented image

1.3. Classification

1.3.1. Deep Learning Models

Proposed system used pre-trained models such as InceptionV3, ResNet50, VGG16, and AlexNet for feature extraction and classification [5-10]. Performance of InceptionV3 is better than other model and achieves higher accuracy.

1.3.2. Convolutional Neural Network:

The presented CNN begins with a convolutional layer containing 64 filters of size (3×3), which are leveraged to extract local features from the input image, as depicted Fig. 3. This is followed by activation function Rectified Linear Unit (ReLU) which introduces non-linearity by converting values that are negative to zero, allowing the system to learn complex data patterns. After convolution, a Max Pooling layer is applied using a (2×2) window to shrink the spatial size by extracting the highest value from each region. This helps in minimizing computational complexity and focusing on the most prominent features. The output is then fed into a Flatten layer, which transforms the 2D feature maps into a 1D vector suitable for fed into fully connected layers. The architecture comprises two dense (fully connected) layers. The initial dense layer includes 256 units and uses ReLU activation to process the input vector. The final dense layer contains a number of unit's equivalent to the number of output classes and uses the softmax activation function to generate probability scores, indicating the model's confidence in predicting each class [11]. Table 1 presents a summary of the proposed CNN model.

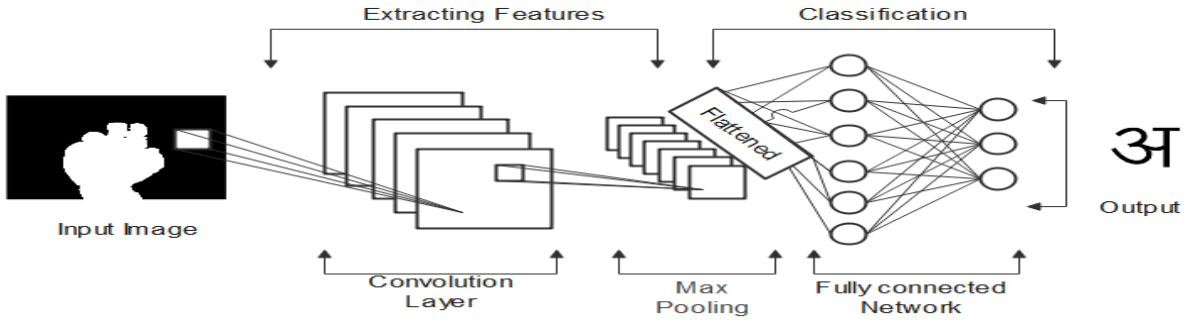

Fig 3: CNN Architecture

Table 1: Structure of Proposed CNN

Layer type	Output Size	Parameters
Conv2d_1	(62,62,64)	1792
Maxpolling2d_1	(31,31,64)	0
Conv2d_2	(29,29,128)	73856
Maxpolling2d_2	(14,14,128)	0
Conv2d_3	(12,12,256)	295168
Maxpolling2d_3	(6,6,256)	0
Flatten	9216	0
Dense_1	256	2359552
Dense_2	47	12079
Total Parameters	2742447	
Trainable Parameters	2742447	
Non-trainable Parameters	0	

1.4.Experimental Setup and Results

We used a 64-bit architecture laptop Windows 10 operating system, 16 GB of RAM with Intel Core i7-9700, and 3.00 GHz with x-64 based processor. The Devnagari sign recognition model is developed using Jupyter notebook with tensorflow, scikit-learn library for machine learning. The dataset is captured using mobile camera. The dataset comprised 4230 images distributed across 47 classes as 90 images of each alphabet. To train and assess the model, we allocated 80% images for training and 20% images testing. The training process extended over 20 to 100 epochs, employing the Adam optimizer alongside the categorical cross-entropy loss function. The Devnagari Sign Language Recognition model's performance was assessed using precision, recall, and F1-score metrics. Table 2 and 3 shows performance evaluation of the proposed methodology and Customized CNN

Table 2: Performance evaluation of the proposed methodology

Deep models	Train Accuracy	Val Accuracy	Precision	Recall	F1-Score	Testing Accuracy	Epochs
InceptionV3	**97.67**	**97.82**	**95**	**95**	**95**	**94.98**	**20**
ResNet50	97.49	97.67	92	92	91	91.63	20
AlexNet	91.88	96.12	89	88	88	88.17	20
VGG16	91.22	91.52	91	92	91	91.51	20

Table 3: Evaluation metrics Customized CNN

Deep models	Train Accuracy	Val Accuracy	Precision	Recall	F1-Score	Testing Accuracy	Epochs
Customized CNN	87.34	89.16	84	82	82	82.19	20
	91.73	94.27	87	86	86	86.14	50
	95.34	96.48	89	88	88	87.93	100
	95.97	**97.07**	**89**	**88**	**89**	**88.41**	**200**

1.4.1. Comparative Analysis with related studies

Table 4: Comparison with related studies

Author	Sign Language	No of Alphabets	Technique Used	Accuracy
Deshpande A.M. [4]	Marathi Sign Language	25	Convolutional Neural Network	99.28%
Das A[12]	American Sign Language	26	Inception V3	90%
Saini B[7]	Indian Sign Language	26	Inception V3	70.54%
			CNN	83.00%
Proposed Inception V3	Devnagari Sign Language	47	Inception V3	94.98%
Proposed CNN	Devnagari Sign Language	47	CNN	88.41%

2. Module 2: Hand Landmark Based DSL recognition system

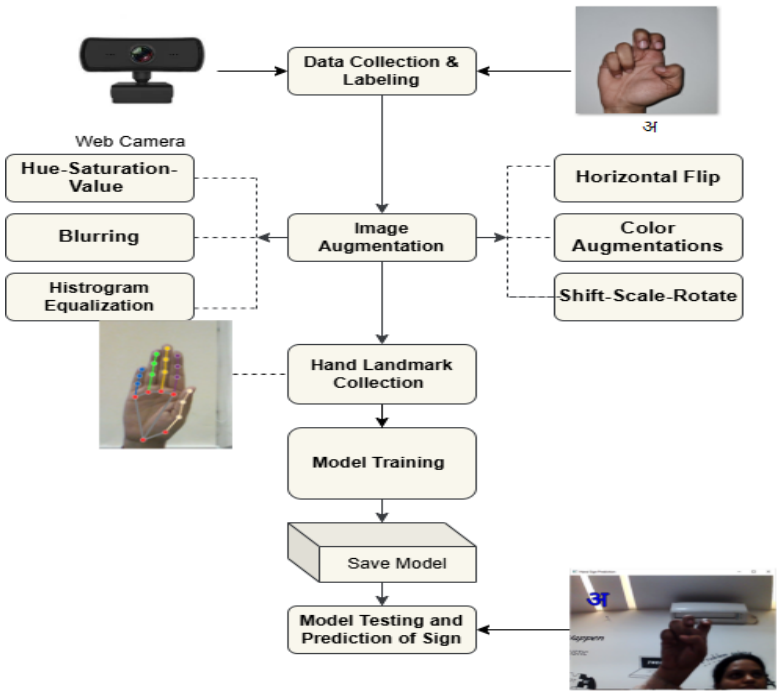

Fig 4: Hand landmark based system

Fig 4 shows flow of hand landmark based system for recognition of 47 Devnagari sign language alphabet. It consist Data collection, Image augmentation then hand landmark collection of augmented images. For model training and testing use different classifier such as SVM, FFNN, Random Forest, LSTM and LSTM with attention layer.

2.1. Data Collection

The dataset includes hand gesture images representing all 47 alphabets of the Devnagari Sign Language, ranging from 'Aa' to 'Dnya', as depicted in Fig. 1. The signs were recorded from individuals of different age groups and genders. To ensure diversity, the images were taken under varied lighting conditions and backgrounds. Augmentation techniques like horizontal flip, blurring, rotation, scaling, contrast, brightness, and color adjustments were applied to increase diversity. In total, the dataset consists of 35,250 images, with 750 samples for each sign class.

2.2. Hand Landmark Collection:

Following image augmentation, hand landmark keypoints were extracted from the images using MediaPipe Hands, an AI-based hand tracking framework. For each detected hand, the X and Y coordinates of key joints—such as the wrist, thumb, and fingers—were recorded, as seen in Fig. 5. The extracted data was organized in a CSV file containing the image path, class label, and coordinates for each landmark. These landmarks include joints like WRIST, THUMB_CMC, THUMB_MCP, THUMB_IP, THUMB_TIP, INDEX_FINGER_MCP, INDEX_FINGER_PIP, INDEX_FINGER_DIP, INDEX_FINGER_TIP, and others listed in Table 5 [14].

Table 5: Hand Landmarks

Hand Landmarks of X-axis	Hand Landmarks of Y-axis
1. WRIST_X	1. WRIST_Y
2. THUMB_CMC_X	2. THUMB_CMC_Y

3. THUMB_MCP_X	3. THUMB_MCP_Y
4. THUMB_IP_X	4. THUMB_IP_Y
5. THUMB_TIP_X	5. THUMB_TIP_Y
6. INDEX_FINGER_MCP_X	6. INDEX_FINGER_MCP_Y
7. INDEX_FINGER_PIP_X	7. INDEX_FINGER_PIP_Y
8. INDEX_FINGER_DIP_X	8. INDEX_FINGER_DIP_Y
9. INDEX_FINGER_TIP_X	9. INDEX_FINGER_TIP_Y
10. MIDDLE_FINGER_MCP_X	10. MIDDLE_FINGER_MCP_Y
11. MIDDLE_FINGER_PIP_X	11. MIDDLE_FINGER_PIP_Y
12. MIDDLE_FINGER_DIP_X	12. MIDDLE_FINGER_DIP_Y
13. MIDDLE_FINGER_TIP_X	13. MIDDLE_FINGER_TIP_Y
14. RING_FINGER_MCP_X	14. RING_FINGER_MCP_Y
15. RING_FINGER_PIP_X	15. RING_FINGER_PIP_Y
16. RING_FINGER_DIP_X	16. RING_FINGER_DIP_Y
17. RING_FINGER_TIP_X	17. RING_FINGER_TIP_Y
18. PINKY_MCP_X	18. PINKY_MCP_Y
19. PINKY_PIP_X	19. PINKY_PIP_Y
20. PINKY_DIP_X	20. PINKY_DIP_Y
21. PINKY_TIP_Xs	21. PINKY_TIP_Y

Fig 5. Hand Keypoints

2.3.Model Training and Testing:

2.3.1. LSTM based RNN

The LSTM cell comprises three primary gates: input, forget, and output. The forget gate filters out unnecessary cell state data, driven by the current input (Xt) and previous output (ht−1). The input gate adds relevant data using a combination of sigmoid and tanh functions to update the cell state. The output gate identifies crucial information from the cell state to produce the final output, which is also passed to the next LSTM cell [15].

Forget Gate

It regulates the elimination of unnecessary information from the prior cell state. The decision is based on the current input, previous hidden state, and passed through a sigmoid activation:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f \quad (1)$$

Where f_t : Forget gate vector, W_f: forget gate's Weight matrix , h_{t-1} : The previous time step's hidden state, b_f : forget gate's Bias vector, σ: Sigmoid activation function

Input Gate

It identifies and stores essential new information. It consists of two steps:

A. Compute the input gate activation:

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i \quad (2)$$

B. Compute the candidate cell state:

$$\widetilde{C_t} = \tanh(W_C[h_{t-1}, x_t] + b_C \quad (3)$$

Cell State Update

The old cell state is updated to by forgetting irrelevant information and incorporating new updates:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (4)$$

Output Gate

Determines the output based on the updated cell state, filtered by a sigmoid gate. The filtered cell state passes through tanh to regulate its range:

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o \quad (5)$$

$$h_t = o_t * \tanh(C_t) \quad (6)$$

2.3.2. Bidirectional LSTM model with a custom attention mechanism:

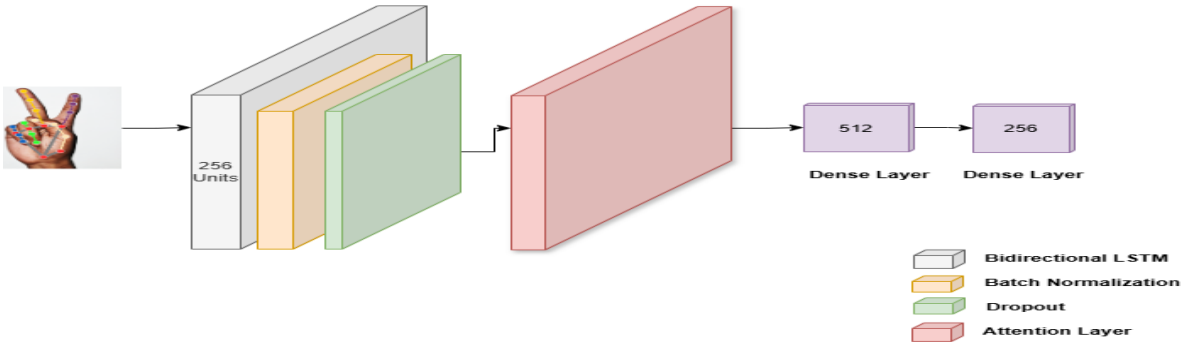

Fig 6. LSTM with Attention Layer

A LSTM based RNN that is specifically designed to remember long-range dependencies and patterns over time. It has internal mechanisms—called gates—that help it retain or forget information as needed.

- Bidirectional LSTM processes the input sequence examined in both directions: forward and backward.
- This dual perspective is crucial for gesture recognition, as understanding a gesture may require knowledge of not merely what has taken place, but also what is about to happen.

By capturing context from both ends, the model can better interpret the overall meaning of the sequence.

A. Input Representation

Input shape: X ∈ ℝ^(N, T, F)
Where:
- N: Number of samples (batch size)
- T = 7: Time steps (from keypoint sequences)
- F = 6: Features per time step (keypoint dimensions)

B. Bidirectional LSTM Layer

Each LSTM processes sequence both forward and backward. Forward LSTM Equations are same as equation 1 to 6. In a Bidirectional LSTM, two LSTMs run in parallel: one forward (from t=1 to T), another backward (from t=T to 1). The outputs from both directions are concatenated as shown in equation 7:

$$h_t^{bi} = [\rightarrow h_t \; ; \; \leftarrow h_t \;] \quad (7)$$

So the output shape is (N,T,512), since 256 units in both directions.

C.Attention Mechanism:

In any sequence, not all time steps are equally important. For instance, some frames in a hand gesture might be more distinctive than others. The attention mechanism is designed to learn which parts of the sequence matter the most for making a prediction.

Here's how it works in this model:

1. Attention Score Calculation: For every time step in the sequence, a small neural network generates a score that reflects the importance of that step as shown in equation 8.
2. Softmax Normalization: These scores are converted into probabilities using the softmax function. This ensures all scores sum up to 1, effectively weighing each time step as shown in equation 9.

3. Weighted Sum: The sequence is then combined into a single vector, where more important steps (based on the attention scores) contribute more as shown in equation 10.

Attention Layer Transformation:

The attention mechanism applies a linear transformation to the LSTM hidden state as follows:

$$et = tanh(Wa \cdot h_t + ba) \quad (8)$$

Where:
- ht: Hidden state from LSTM at time step t
- Wa: Weight matrix for attention
- ba: Bias term for attention layer
- et: Raw attention score for time step t

2. Softmax Attention Weights

Softmax is applied to convert attention scores into attention weights:

$$\alpha_t = \frac{exp(et)}{\sum_{k=1}^{T} exp(et)} \quad (9)$$

Where:
- α_t: Attention weight at time step t
- et: Attention score at time t

3. Context Vector

The context vector is the weighted sum of the hidden states:

$$c = \sum_{t=1}^{T} \alpha_t \cdot h_t \quad (10)$$

Where:
- c: Context vector
- α_t: Attention weight at time step t
- h_t: Hidden state from LSTM at time t

Dense Layers:

After the attention mechanism condenses the sequence into a meaningful vector, this information is passed through a series of fully connected (dense) layers. These layers learn complex combinations of features that help distinguish between different classes.

Each layer is followed by:

- Batch Normalization: Stabilizes and speeds up training.
- Dropout: Prevents overfitting by randomly ignoring some neurons during training.

Output Layer: Classification

The final layer is a softmax classifier that outputs the probability distribution across all gesture classes. The class with the highest probability is chosen as the model's prediction [16].

2.4. Experimental results

The experimental setup was conducted on a 64-bit Windows 10 laptop equipped with 8 GB RAM and an Intel Core i3-6100U CPU (2.30 GHz). The model for Devnagari character recognition was developed in Jupyter Notebook using MediaPipe, TensorFlow, and scikit-learn. The dataset included 35,250 images across 47 sign classes, with 750 samples each. A training-to-testing ratio of 80:20 was used. Performance of model compared using different classification techniques like SVM, FFNN, Random Forest, LSTM and LSTM with attention layer as depicted in Fig 7. The model underwent training for 200 epochs using the Adam optimizer and categorical cross-entropy loss. The effectiveness of the Devnagari alphabet recognition system is assessed through the classification report & confusion matrix. The classification report includes key metrics like precision, recall, F1-score, and support for each character, which demonstrate the model's accuracy in

identifying characters. Proposed system achieves higher accuracy using LSTM with AM as 98.05%. Fig 8 and 9 shows predicted output of different sign gesture with different lighting condition and background in static and real time mode. Table 7 shows comparative analysis with related studies.

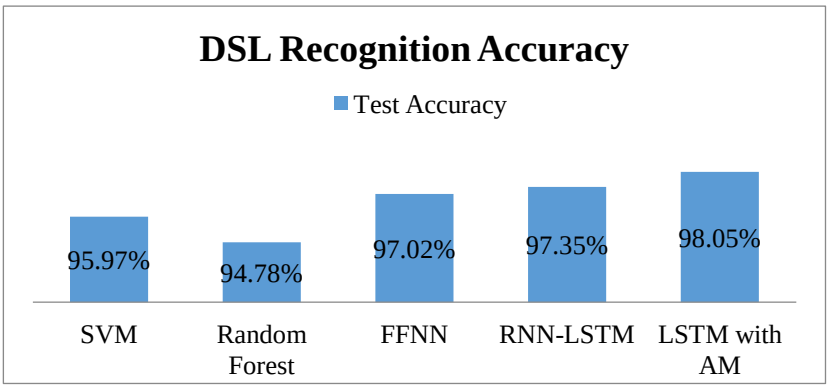

Fig 7: Result of Different classifier

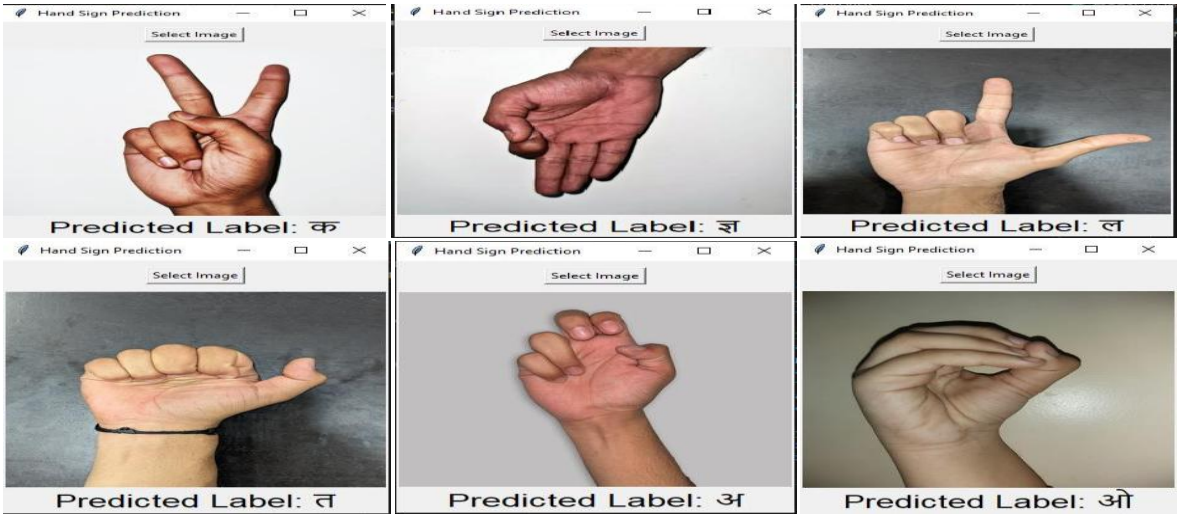

Fig 8. Predicted output with different lighting condition and background

Fig 9. Real time recognition of DSL signs

Table 7: Comparative Analysis with related studies

Author	Sign Language	No of Alphabets	Technique Used	Accuracy
Avola, D [15]	American Sign Language	26	Leap Motion Controller, Recurrent neural network	96%
Chong & Lee [17]	American Sign Language	26	Support vector machine and a Deep neural network	80.30% and 93.81%, respectively
M. A. Ihsan [18]	Medical signs(English)	30	MobileNetV2 BiLSTM with AM	95.83%
Ezhil Tharsan S [19]	Indian Sign Language	26	LSTM-RNN and Mediapipe	93.8%
Proposed System	Devnagari Sign Language	47	LSTM-RNN	97.35%
Proposed System	Devnagari Sign Language	47	LSTM-RNN with AM	98.05%

3. MODULE 3: Barakhadi Recognition of DSL

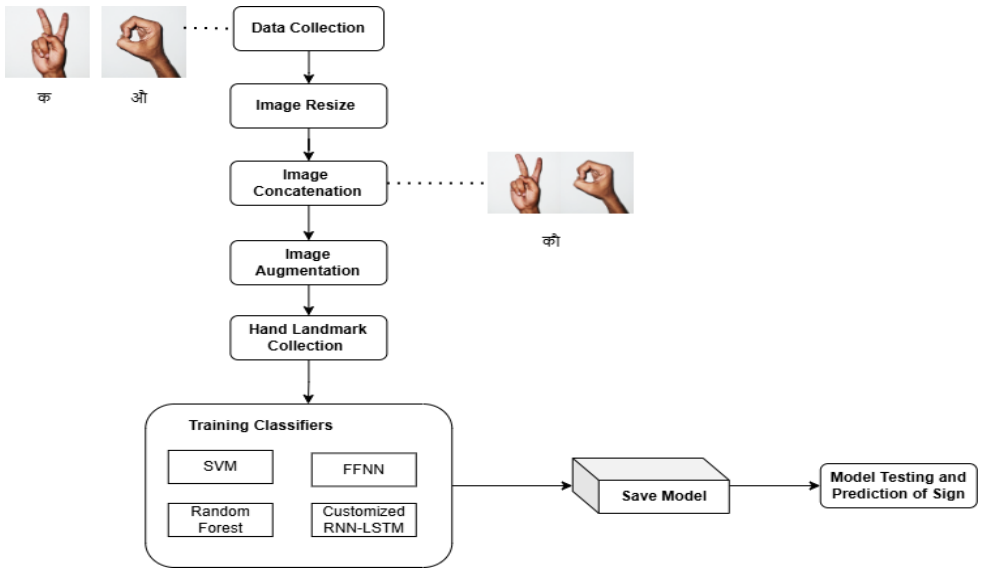

Fig 10. Proposed System for Barakhadi Recognition

3.1. Data Collection and Augmentation

A dataset of 47 Devnagari Sign Language alphabets including vowels and consonants collected from 15 individuals using a mobile camera under varied conditions and background. To create the complete 408 Barakhadi set, 34 consonants were combined with 12 vowels as shown in fig 11. Augmentation techniques like horizontal flip, blurring, rotation, scaling, contrast, brightness, and color adjustments were applied to increase diversity. The final dataset contains 306,000 images, with 750 samples per class.

3.2. Hand Landmark Extraction

Using MediaPipe Hands, keypoint coordinates (wrist and finger joints) were retrieved from the augmented images. These were recorded in a CSV file, together with class labels for further model training. Details are provided in Table 8.

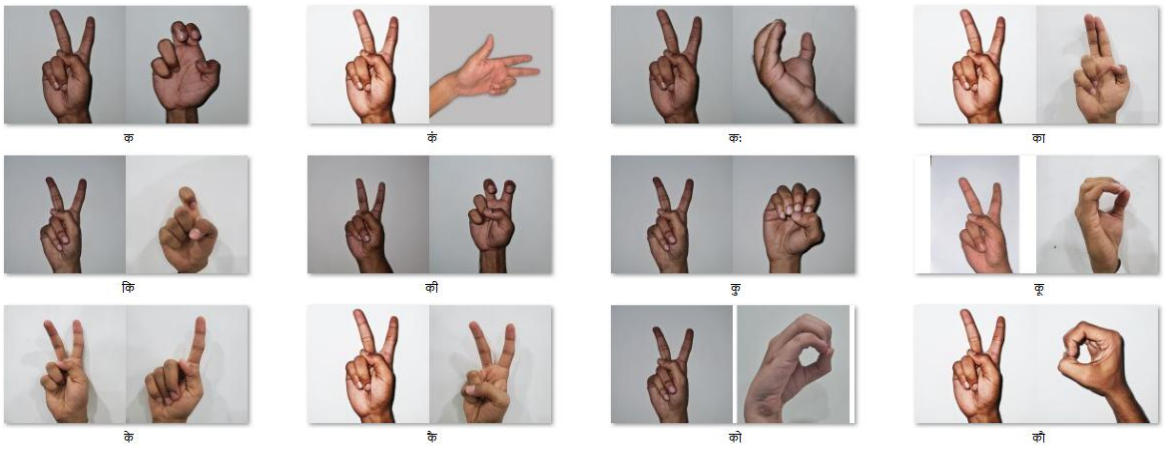

Fig 11. Sample of alphabet 'क' Concatenation with 12 vowels

Table 8: Hand Landmarks

Fingers	Hand Landmarks	Landmarks of 1st Hand Image (X and Y coordinates)	Landmarks of 2nd Hand Image (X and Y coordinates)
WRIST	-	2	2
THUMB	CMC,MCP,IP,TIP	8	8
INDEX	MCP,PIP,TIP,DIP	8	8

MIDDLE	MCP,PIP,TIP,DIP	8	8
RING	MCP,PIP,TIP,DIP	8	8
PINKY	MCP,PIP,TIP,DIP	8	8
Total Landmarks		84	

3.3.Normalization

To ensure consistency in data scale, decimal scaling normalization was applied. This technique adjusts each value by dividing it by a power of 10, based on the largest value in the dataset. It preserves the relative differences between values while bringing them into a comparable range, typically between 0 and 1.

3.4. Classification

3.4.1. Feed Forward Neural Network (FFNN)

A fully connected neural network was implemented for classifying the dataset .The input layer accepts features matching the training data dimensions. ReLU activation introduces non-linearity across multiple dense layers, which gradually reduce in size (512, 256, 128 units) to learn hierarchical features. Dropout layers are added after each dense layer to prevent overfitting. The final output layer uses softmax activation to predict class probabilities across all 408 Barakhadi categories [20].

3.4.2. Customized RNN with LSTM

A customized Recurrent Neural Network (RNN) using Long Short-Term Memory (LSTM) layers was developed for recognizing Barakhadi sequences. LSTMs are ideal for sequential data, as they effectively capture long- and short-term patterns while overcoming limitations like vanishing gradients in traditional RNNs. Their internal gating mechanisms allow better handling of temporal dependencies in the sign language data. The model is composed of multiple layers, starting with an LSTM layer containing 256 units, which outputs the full sequence to retain temporal dependencies in the data. This is followed by a batch normalization layer to stabilize learning and accelerate training. A dropout layer with a 0.5 rate is included to prevent overfitting. The next LSTM layer, with 128 units, processes the sequence and outputs a fixed-size vector summarizing the temporal features. This vector is passed through two dense (fully connected) layers: the first with 64 units using ReLU activation, and the second with 408 units corresponding to the number of gesture classes, using a softmax activation to generate class probabilities. Additional dropout layers are applied after the dense layers to improve generalization [15]. The architecture of customized LSTM is shown in fig 12.

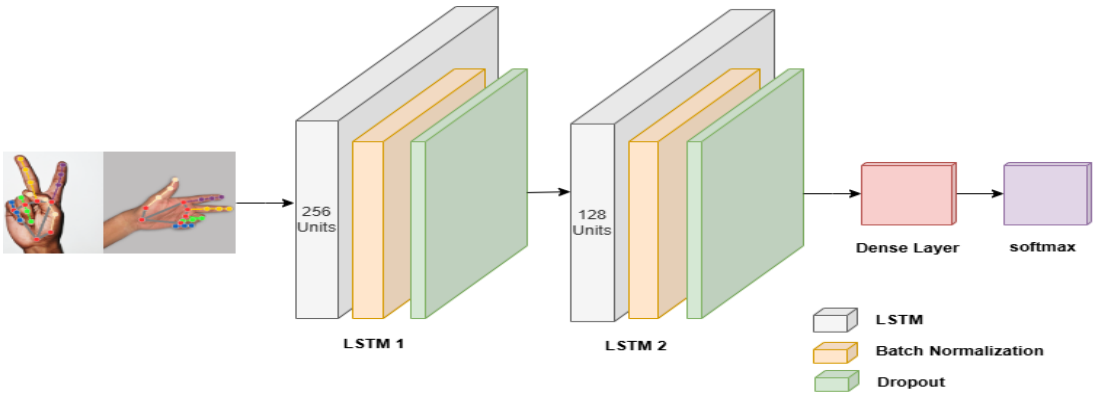

Fig 12.Achitecture of Customized LSTM

3.5. Experimental results

The experimental setup was carried out on a 64-bit Windows 10 laptop equipped with 8 GB RAM and an Intel Core i3-6100U processor (2.30 GHz). The Barakhadi recognition system for Devnagari Sign Language was developed in a Jupyter Notebook using TensorFlow, Mediapipe and scikit-learn. The dataset comprised 30,600 images across 408 classes as 750 images per class. The data was split with 80% for training and 20% for testing. The model was trained over 50 epochs using a batch size of 32, employing sparse categorical cross-entropy as the loss function. Multiple optimizers such as Adam, RMSprop, SGD, and Nadam were tested, with all showing early convergence as shown in fig 14. Performance metrics including accuracy, precision, recall, and F1-score were used for evaluation.

A customized RNN-LSTM model achieved the highest accuracy of 99.62%, outperforming FFNN (97.14%), SVM (97.51%), and Random Forest (93.66%) as shown in Fig 13. This highlights the strength of sequence-based models in recognizing temporal patterns in sign language.

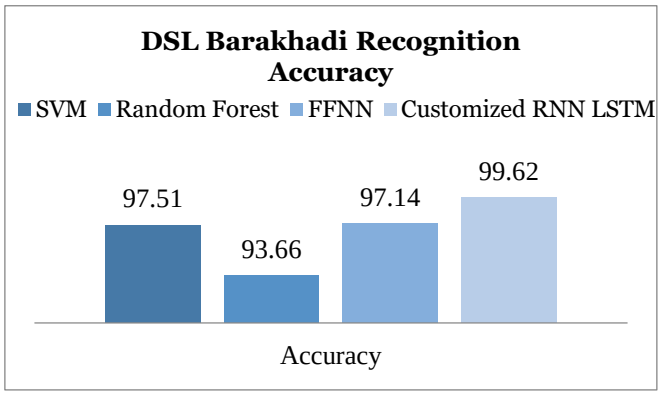

Fig.13. Performance Comparison of Different Classifiers

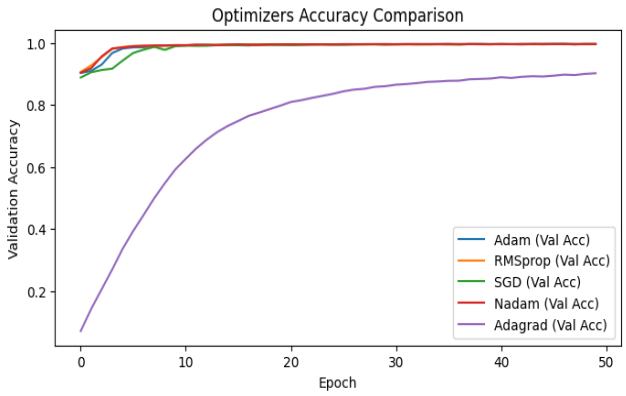

Fig 14.Different Optimizers Accuracy

CONCLUSION

This research presents a comprehensive and innovative approach to enhancing the recognition of Devnagari Sign Language (DSL). By constructing a diverse and balanced dataset that captures variations across age, gender, lighting conditions, and backgrounds, we ensured a robust foundation for effective gesture recognition. The integration of vowel signs with consonants allowed for the creation of 408 unique gesture classes, addressing the complexity of the Devnagari script and enabling comprehensive linguistic coverage.

Through extensive experimentation, the study evaluated the performance of various pre-trained models, including InceptionV3, ResNet50, VGG16, and AlexNet, alongside a custom CNN architecture for 47 Devnagari alphabets. InceptionV3 outperforming others in classification tasks, emerged as the most accurate and reliable model.

Furthermore, advanced image augmentation techniques, coupled with hand landmark extraction using MediaPipe, facilitated reliable recognition even under variable environmental conditions. In the comparative analysis of classification methods—SVM, Random Forest, FFNN, LSTM-based RNN and LSTM with AM—the LSTM with AM model demonstrated superior performance, particularly in recognizing DSL gestures in both static and real-time settings. The proposed LSTM with AM model achieved a notable accuracy of 98.05%, confirming its effectiveness in learning sequential patterns inherent in sign language gestures. The dataset is expanded to include barakhadi of Devnagari script. Combination of 12 vowel signs with 34 consonants created 408 unique gesture classes of barakhadi. Among the classifiers examined in the study, the customized RNN-LSTM architecture achieved the highest accuracy of 99.61%.

Future research could expand on this methodology to include real-time recognition of Devnagari words, and complete sentences. This extension promises to further enhance the system's utility and effectiveness in supporting inclusive communication for the hearing-impaired community in India. Future work could focus on expanding the dataset to include dynamic gestures and real-time recognition of all 408 barakhadi alphabets. To bridging communication gaps integrate this system into portable devices or mobile applications could make DSL recognition accessible to a wider audience.

ORGANISATION OF THE THESIS

TABLE OF CONTENTS

Chapter 1: Introduction
1.1 Overview
1.2 Motivation of Research
1.3 Thesis Organisation

Chapter 2: Literature Review
2.1 Background
2.2 Literature Review
2.3 Identification of Research Gaps

Chapter 3: Problem Statement, Objectives
3.1 Problem Statement
3.2 Objectives
3.3 Contributions

Chapter 4: Proposed Methodology
Module 1: Image Based DSL Alphabet Recognition
4.1 Introduction
4.2 Dataset Collection & Pre-processing
4.3 Deep Learning Approaches
4.4 Experimental Setup
4.5 Results and Discussion
4.6 Comparative Analysis
4.7 Summary

Chapter 5: Proposed Methodology
Module 2: Hand Landmark based DSL Alphabet Recognition
5.1 Introduction
5.2 Dataset Collection
5.3 Hand Landmark Collection
5.4 Machine Learning Approaches
5.5 Experimental Setup
5.6 Results and Discussion
5.7 Comparative Analysis
5.8 Summary

Chapter 6: Proposed Methodology
Module 3: Hand Landmark based DSL Barakhadi Recognition
6.1 Introduction
6.2 Dataset Collection
6.3 Hand Landmark Collection
6.4 Machine Learning Approaches
6.5 Experimental Setup
6.6 Results and Discussion
6.7 Comparative Analysis
6.8 Summary

Chapter 7: Conclusion
7.1. Conclusion
7.2 Scope of Future Work

References

References

[1] Gaikwad. R.,; Admuthe, L. (2024).; Sign Language Recognition of Words and Sentence Prediction Using Lstm and Nlp. Journal of Theoretical and Applied Information Technology, 102(4), 1541–1548.

[2] Pansare, J., & Ingle, M. (2018). A Real-Time Devnagari Sign Language Recognizer (α-DSLR) for Devnagari Script. In Lecture Notes in Networks and Systems (Vol. 18, pp. 75–84). Springer. https://doi.org/10.1007/978-981-10-6916-1_8

[3] Shazid wahid khandakhani,; Sachikanta Dash,; Sasmita Padhy,; Rabinarayan Panda.; (2024). Implementation of Deep learning Methods to Marathi Hand Written Characters and its Pattern Recognition by Using Generative AI. Journal of Computational Analysis and Applications (JoCAAA), 33(07), 979–994.

[4] Deshpande, A.M., Kalbhor, S.R. (2020). Video-Based Marathi Sign Language Recognition and Text Conversion Using Convolutional Neural Network, Lecture Notes in Electrical Engineering, vol 569. Springer, Singapore.

[5] Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference oncomputer vision and pattern recognition; 2016. p. 2818–26.

[6] Ewe, E.L.R.; Lee, C.P.; Kwek, L.C.; Lim, K.M. Hand Gesture Recognition via Lightweight VGG16 and Ensemble Classifier. Appl. Sci. 2022, 12, 7643. https://doi.org/10.3390/app12157643

[7] Saini B, Venkatesh D, Chaudhari N, et al. (2023) A comparative analysis of Indian sign language recognition using deep learning models. Forum for Linguistic Studies 5(1): 197–222. DOI: 10.18063/fls.v5i1.1617

[8] Rathi P, Gupta RK, Agarwal S, and Shukla A (2020) Sign language recognition using ResNet50 deep neural network architecture. Social Science Research Network. DOI: 10.2139/ssrn.3545064.2

[9] He, K., Zhang, X., Ren, S. and Sun, J. (2016) Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770-778.https://doi.org/10.1109/CVPR.2016.90

[10] Krizhevsky, A.; Sutskever, I.; Hinton, G.E.Imagenet classification with deep Convolutional neural networks. In Proceedings of the Twenty-Sixth Annual Conference on Neural Information Processing Systems. Lake Tahoe, NY, USA, 3–8 December 2012; pp. 1097–1105.

[11] Wadhawan, A., & Kumar, P. (2020). Deep learning-based sign language recognition system for static signs. Neural Computing and Applications, 32(12), 7957–7968.

[12] Das, A., Gawde, S., Suratwala, K., & Kalbande, D. (2018). Sign Language Recognition Using Deep Learning on Custom Processed Static Gesture Images. 2018 International Conference on Smart City and Emerging Technology (ICSCET). doi:10.1109/icscet.2018.8537248.

[13] A. Hussain, S. U. Amin, M. Fayaz and S. Seo, "An efficient and robust hand gesture recognition system of sign language employing finetuned inception-v3 and efficientnet-b0 network," Computer Systems Science and Engineering, vol. 46, no.3, pp. 3509–3525, 2023.

[14] Indriani, Harris, M., & Agoes, A. S. (2021). Applying Hand Gesture Recognition for User Guide Application Using MediaPipe. Proceedings of the 2nd International Seminar of Science and Applied Technology (ISSAT 2021), 207(Issat), 101–108. https://doi.org/10.2991/aer.k.211106.017

[15] Avola, D., Bernardi, M., Cinque, L., Foresti, G. L., & Massaroni, C. (2019). Exploiting Recurrent Neural Networks and Leap Motion Controller for the Recognition of Sign Language and Semaphoric

Hand Gestures. IEEE Transactions on Multimedia, 21(1), 234–245. https://doi.org/10.1109/TMM.2018.2856094

[16] W. Jian et al., "SA-Bi-LSTM: Self Attention With Bi-Directional LSTM-Based Intelligent Model for Accurate Fake News Detection to Ensured Information Integrity on Social Media Platforms," in IEEE Access, vol. 12, pp. 48436-48452, 2024, doi: 10.1109/ACCESS.2024.3382832

[17] Chong, T.-W.; Lee, B.-G. American Sign Language Recognition Using Leap Motion Controller with Machine Learning Approach. Sensors 2018, 18, 3554. https://doi.org/10.3390/s18103554

[18] M. A. Ihsan, A. F. Eram, L. Nahar and M. A. Kadir, "MediSign: An Attention-Based CNN-BiLSTM Approach of Classifying Word Level Signs for Patient-Doctor Interaction in Hearing Impaired Community," in IEEE Access, vol. 12, pp. 33803-33815, 2024, doi: 10.1109/ACCESS.2024.3370684.

[19] Ezhil Tharsan S., Dharshan S., Dinesh G., Saraswathi S. . Real Time Indian Sign Language Detection using LSTM and Keypoint Extraction. International Journal of Computer Applications. 184, 21 (Jul 2022), 1-7. DOI=10.5120/ijca2022922235

[20] P. Suresh Kumar, H.S. Behera, Anisha Kumari K, Janmenjoy Nayak, Bighnaraj Naik, Advancement from neural networks to deep learning in software effort estimation: Perspective of two decades, Computer Science Review, Volume 38, 2020,100288, ISSN 1574-0137, https://doi.org/10.1016/j.cosrev.2020.100288.

List of Publications and Conference Attended

1. **Journals:**
    - 2 Journal paper published (Scopus Indexed+ UGC Care I).
2. **Conference Papers:**
    - 1 Conference paper published (Springer-Scopus Indexed).
    - 1 Conference paper presented. (Springer :Publication in Process)
3. **Intellectual Property:**
    - 1 South Africa Patent granted.

Journals:

1) Deepali R. Naglot, Deepa S. Deshpande, "**A Machine Learning Approaches for Barakhadi Recognition of Devnagari Sign Language**", Journal of Information Systems Engineering and Management, 2025,10(26s) e-ISSN:2468-4376.https://www.jisem-journal.com/index.php/journal/article/view/4251
2) Deepali R. Naglot and Deepa Deshpande, "**Deep Learning Models for Devnagari Sign Language Recognition**" , International Journal of Information Technology and Electrical Engineering (ITEE), 13 (6), pp. 01-12, DEC 2024 ,ISSN: 2306-708X , **http://www.iteejournal.org/v13no6dec24_pdf1.pdf**

Conference Papers:

1) Naglot, D.R., Deshpande, D.S. (2025). Devnagari Sign Language Recognition for Deaf and Mute Individuals Using Convolutional Neural Network. ICT Analysis and Applications. ICT4SD 2024. Lecture Notes in Networks and Systems, vol 1161. Springer, Singapore. https://doi.org/10.1007/978-981-97-8602-2_3
2) Presented: "Devnagari Sign Language Recognition with LSTM-based RNN" in PCEMS 2024, the 3rd International Conference on the Paradigm Shifts in Communication, Embedded systems, Machine learning and Signal processing , Visvesvaraya National Institute of Technology Nagpur, Maharashtra, INDIA

Intellectual Property:

1. South Africa Patent granted – "A SYSTEM AND A METHOD FOR DEVNAGARI SIGN LANGUAGE RECOGNITION", Application number: 2022/06226, Grant date: 2022/08/31